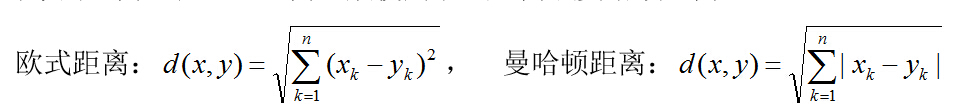
**KNN**

**1、概念**：监督学习方法，给定测试样本，基于某种距离度量找出训练集中与其最靠近的K个样本，然后基于这K个样本的信息确定其类别。

优势：1）通过计算对象间距离来作为各个对象之间的非相似性指标，避免了对象之间的匹配问题2）KNN通过依据k个对象中占优的类别进行决策，而不是单一的对象类别决策。



**2、算法流程**：

1）计算测试数据与各个训练数据之间的距离；

2）按照距离的递增关系进行排序；

3）选取距离最小的K个点；

4）确定前K个点所在类别的出现频率；

5）返回前K个点中出现频率最高的类别作为测试数据的预测分类。

**3、k值选择**

K较小：近似误差小，估计误差大，对近邻点敏感，模型复杂，容易过拟合；

K较大：近似误差大，估计误差小，较远实例也会起作用使预测错误，k大意味着模型简单

问题与答案

1. **推荐系统的算法中最近邻和矩阵分解各自适用场景**
2. **KNN中是如何使用KD树的？**

当样本数量大时，线性扫描搜索计算量大，因此引入kd树这种结构存储训练数据，以减少计算距离的次数。

**kd树构造**：(依次选择坐标轴对空间进行划分，选择训练数据在选定坐标轴上的中位数进行切分，可构造平衡kd树，但平衡kd树未必是最优的)

样本 ,k维特征空间，

1. 开始：构造根节点

选取坐标轴为x(1),选取切分点为样本x(1)坐标的中位数，切分超平面为垂直于x(1)轴并通过切分点的超平面，根节点为坐落在切分超平面上的样本点（即x(1)坐标值为切分点），根节点左子区域为x(1)<切分点，右子区域为大于。深度为1。

1. 重复：构造深度为j的节点，选择第x(l)坐标轴， ，步骤如上

3）停止：左右子区域没有实例

**kd树搜索**：（寻找最近邻，k近邻类似）

1）寻找目标点x所在的叶节点：从根节点出发，向下搜索，直到子节点为叶节点

2）此叶节点为“当前最近点”

3）递归向上回退，每个节点进行以下操作：

* 若该节点保存的实例点比“当前最近点”更近，则该节点为“当前最近点”
* 当前最近点一定存在于该节点的一个子区域内，检查该节点父节点的另一个子节点对应的区域内是否有更近的点。另一子节点对应的区域与以目标点为中心、以当前最近点为半径的超球体相交，则在另一子节点可能存在距离最近点，移动到另一子节点，继续搜索。若不相交，则向上回退
* 当回退到根节点，则“当前最近点”为距离目标点x的最近邻点。

1. **如何对于N个样本，每个样本为D维向量，采用欧式距离使用KNN做类预测。1).给出预测时间复杂度。2).当N很大时，有哪些方法可以降低复杂度？3).k取值的大小对预测方差和偏差有何影响？**

1）

2）kd树（计算复杂度为,适合N远大于D的k近邻搜索，若N与D接近，效率下降接近线性扫描）

3）k小，偏差小，方差大，易受噪点的干扰

k大，偏差大，方差小

[**http://blog.csdn.net/dobests/article/details/48580899**](http://blog.csdn.net/dobests/article/details/48580899)

1. **在k-means或kNN，我们是用欧氏距离来计算最近的邻居之间的距离。为什么不用曼哈顿距离？**

我们不用曼哈顿距离，因为它只计算水平或垂直距离，有维度的限制。另一方面，欧式距离可用于任何空间的距离计算问题。因为，数据点可以存在于任何空间，欧氏距离是更可行的选择。例如：想象一下国际象棋棋盘，象或车所做的移动是由曼哈顿距离计算的，因为它们是在各自的水平和垂直方向的运动。